

Embedded Systems

Applications in Imaging and
Communication



Memory

- Location
 - CPU
 - Internal
 - External
- Capacity
 - Word size
 - The natural unit of organisation
 - Number of words
 - or Bytes



Access Methods

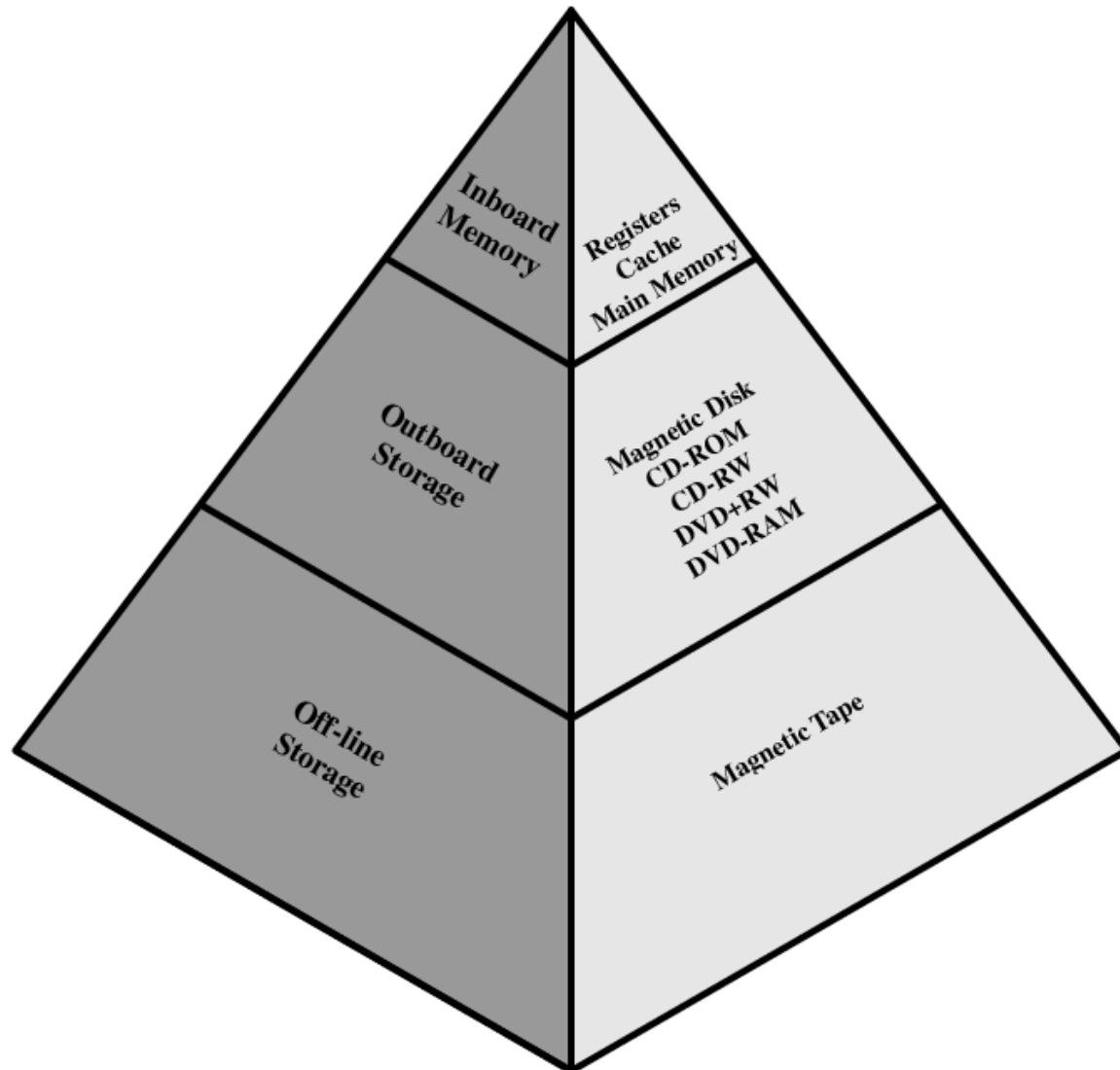
- Sequential
 - Start at the beginning and read through in order
 - Access time depends on location of data and previous location e.g. Tape
- Direct
 - Individual blocks have unique address
 - Access is by jumping to vicinity plus sequential search



Memory Hierarchy

- Registers
 - In CPU
- Internal or Main memory
 - May include one or more levels of cache
 - RAM
- External memory
 - Backing store

Memory Hierarchy - Diagram





Performance

- Access time
 - Time between presenting the address and getting the valid data
- Memory Cycle time
 - Time may be required for the memory to “recover” before next access
 - Cycle time is access + recovery
- Transfer Rate
 - Rate at which data can be moved



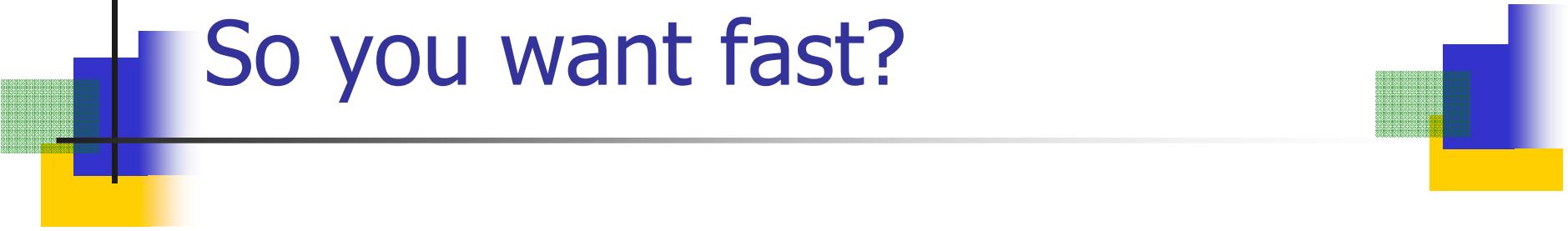
The Bottom Line

- How big?
 - Capacity
- How fast?
 - Access time
- How expensive?
 - Cost/MB



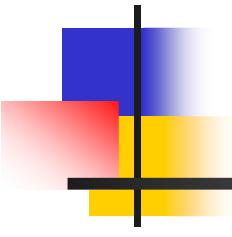
Hierarchy List

- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk
- Optical
- Tape



So you want fast?

- It is possible to build a computer which uses only RAM or Cache
- This would be very fast
- This would cost a very large amount



Cache



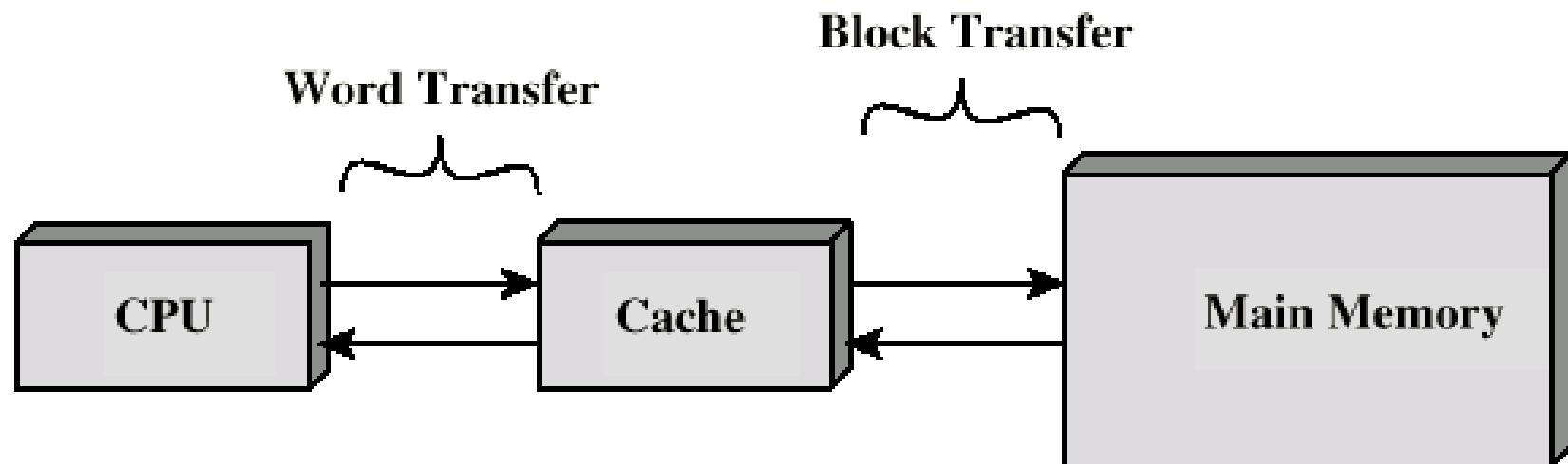
Principle behind Cache Memory

■ Locality of Reference

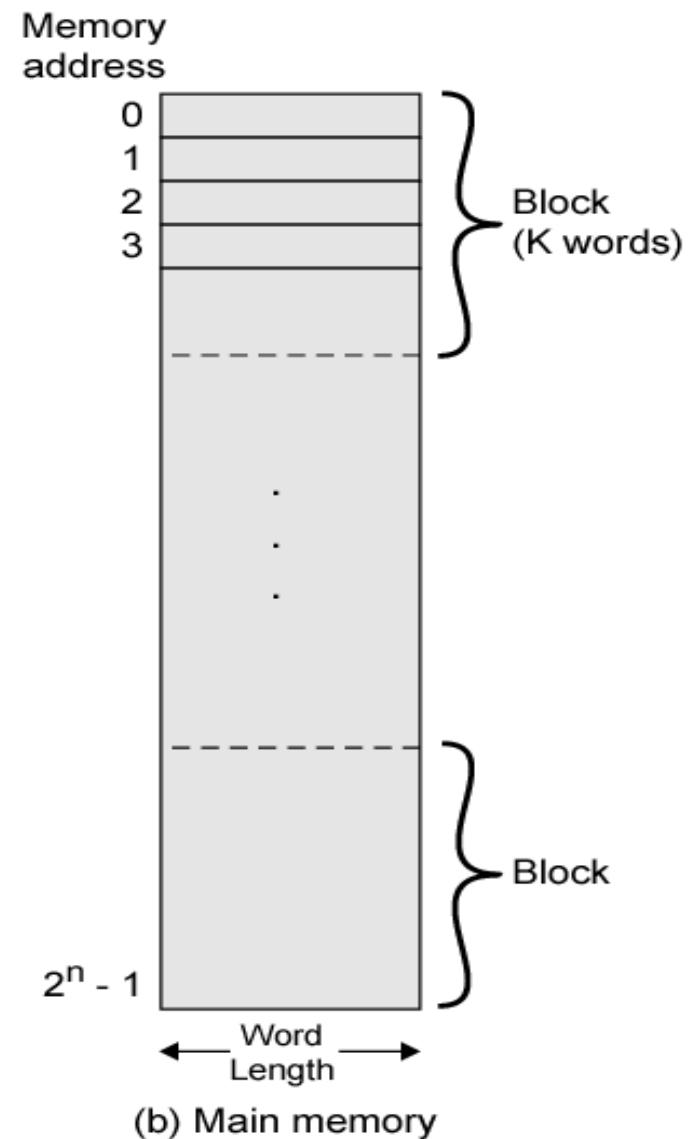
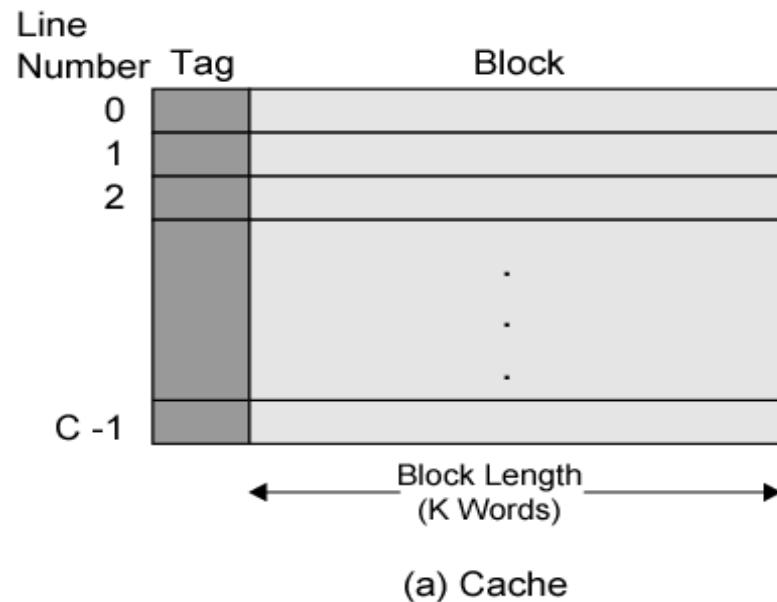
- During the course of the execution of a program, memory references tend to cluster e.g. Loops
- Temporal locality - a recently referenced memory location is likely to be referenced again
- Spatial Locality - a neighbor of a recently referenced memory location is likely to be referenced

Cache

- Small amount of fast memory
- Sits between normal main memory and CPU
- Located on CPU chip or module



Cache/Main Memory Structure

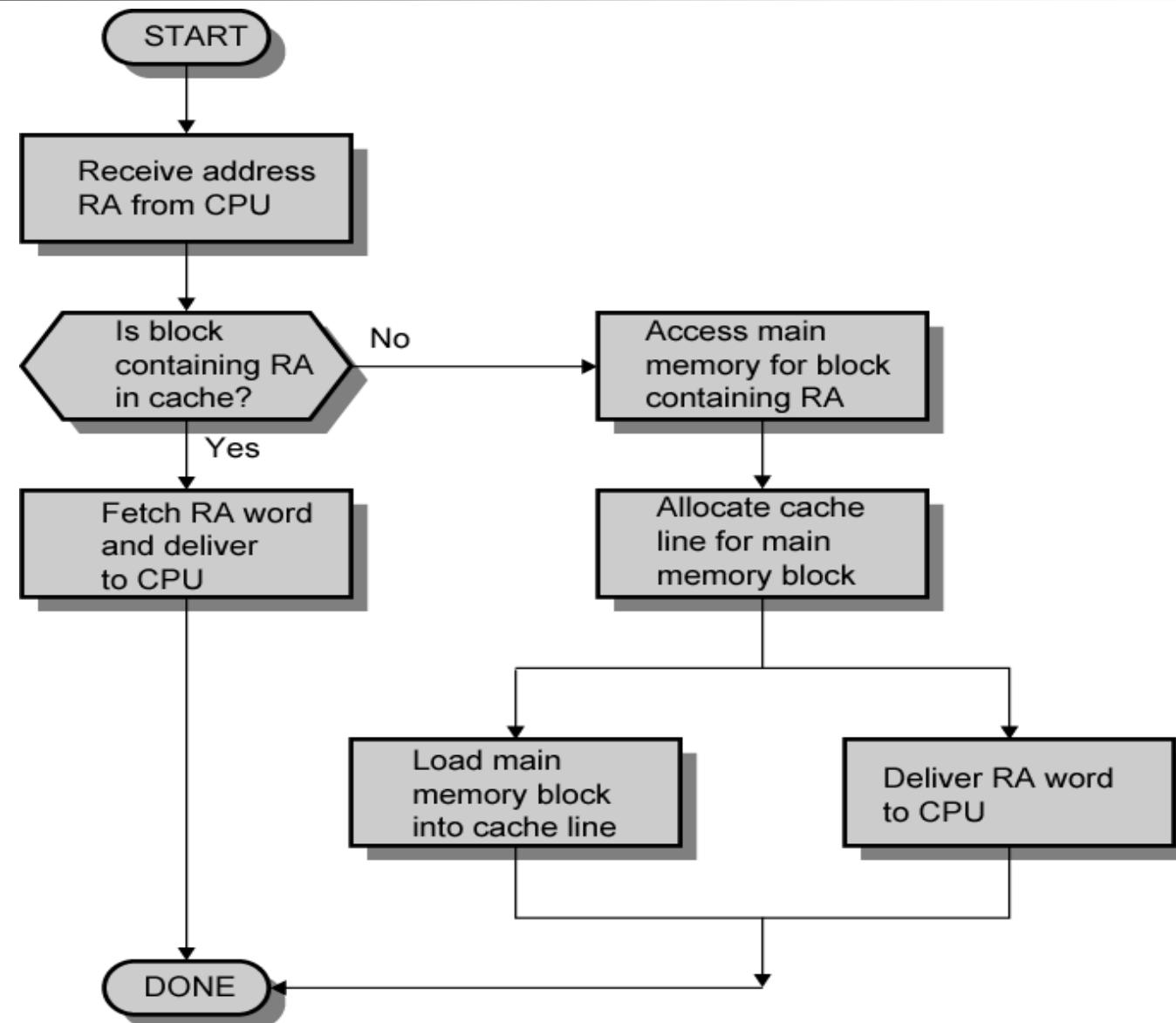




Cache operation – Overview

- Small amount of fast memory
- Sits between normal main memory and CPU
- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast)
- If not present, read required block from main memory to cache
- Then deliver from cache to CPU
- Cache includes tags to identify which block of main memory is in each cache slot

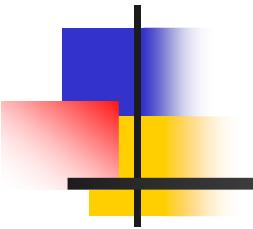
Cache Operation





Cache Design

- Size
- Mapping Function
- Replacement Algorithm
- Write Policy
- Block Size
- Number of Caches



Memory Mapping

- Direct
- Associative
- Set Associative

Mapping Function

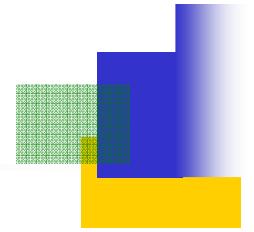
- Cache of 64 kByte
- Cache block of 4 bytes
 - i.e. cache is 16k (2^{14}) lines of 4 bytes
- 16 MBytes main memory
- 24 bit address
 - ($2^{24}=16M$)



Direct Mapping

- Each block of main memory maps to only one cache line
 - i.e. if a block is in cache, it must be in one specific place
- Address is in two parts
- Least Significant w bits identify unique word
- Most Significant s bits specify one memory block
- The MSBs are split into a cache line field r and a tag of $s-r$ (most significant)

Direct Mapping



Tag s-r	Line or Slot r	Word w
8	14	2

- 24 bit address
- 2 bit word identifier (4 byte block)
- 22 bit block identifier
 - 8 bit tag (=22-14)
 - 14 bit slot or line
- No two blocks in the same line have the same Tag field
- Check contents of cache by finding line and checking Tag

Direct Mapping

- Cache line Main Memory blocks held
 - 0 0, m, 2m, 3m...2s-m
 - 1 1,m+1, 2m+1...2s-m+1
 - m-1 m-1, 2m-1,3m-1...2s-1



Direct Mapping pros & cons

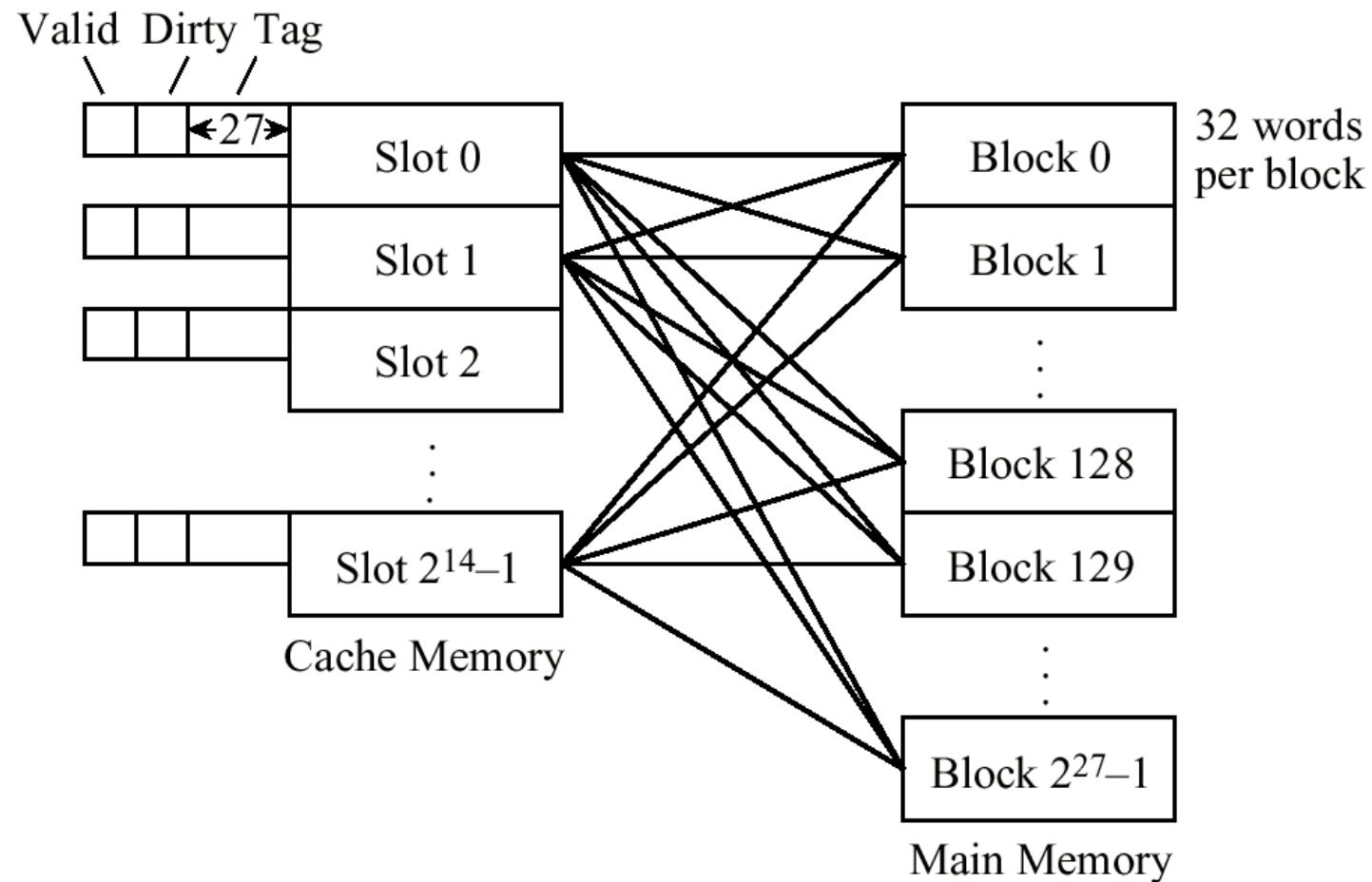
- Simple
- Inexpensive
- Fixed location for given block
 - If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high



Associative Mapping

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive

Associative Mapping





Associative Mapping

- Consider how an access to memory location $(A035F014)_{16}$ is mapped to the cache for a 2^{32} word memory. The memory is divided into 2^{27} blocks of $2^5 = 32$ words per block, and the cache consists of 2^{14} slots:

Associative Mapping

- Consider how an access to memory location $(A035F014)_{16}$ is mapped to the cache for a 2^{32} word memory. The memory is divided into 2^{27} blocks of $2^5 = 32$ words per block, and the cache consists of 2^{14} slots:

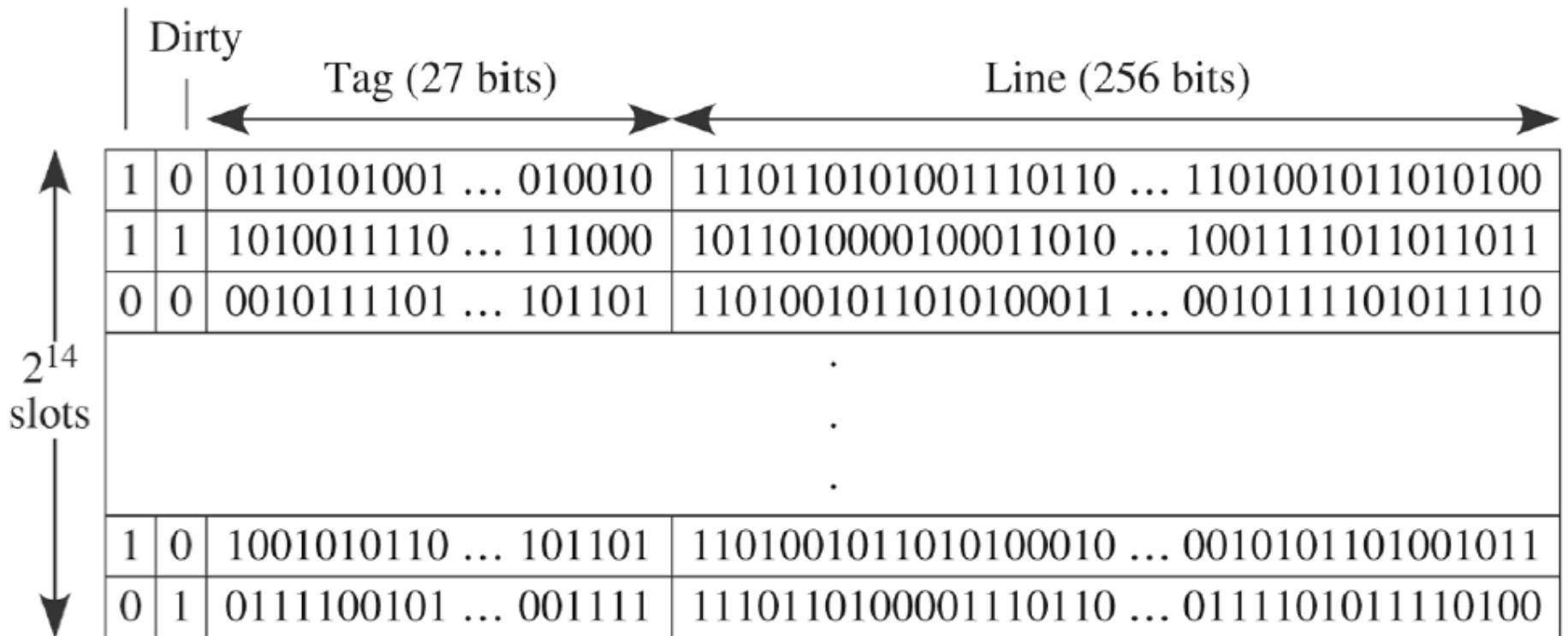
Tag	Word
27 bits	5 bits

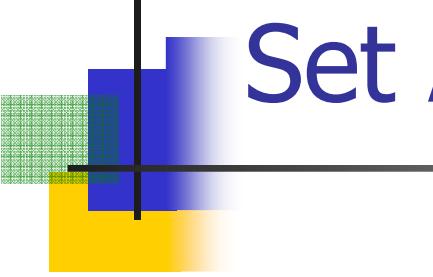
- If the addressed word is in the cache, it will be found in word $(14)_{16}$ of a slot that has tag $(501AF80)_{16}$, which is made up of the 27 most significant bits of the address. If the addressed word is not in the cache, then the block corresponding to tag field $(501AF80)_{16}$ is brought into an available slot in the cache from the main memory, and the memory reference is then satisfied from the cache.

Tag	Word
1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0	1 0 1 0 0

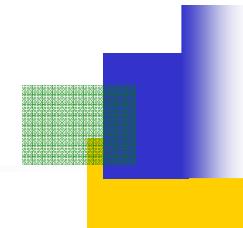
Associative Mapping

Valid





Set Associative Mapping



- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set
 - e.g. Block B can be in any line of set i
- e.g. 2 lines per set
 - 2 way associative mapping
 - A given block can be in one of 2 lines in only one set

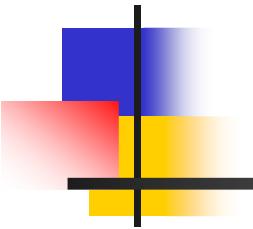
Set Associative Mapping

Tag 12 bit

Set 15 bit

Word
5 bit

- Use set field to determine cache set to look in
- Compare tag field to see if we have a hit



Block Replacement Methods



Replacement Algorithms

- Direct Mapping
 - No choice
 - Each block only maps to one line
 - Replace that line



Replacement Algorithms

- Associative and Set Associative Mapping
 - Least Recently used (LRU)
 - e.g. in 2 way set associative
 - which of the 2 block is LRU?
 - First in first out (FIFO)
 - replace block that has been in cache longest
 - Least frequently used
 - replace block which has had fewest hits



Write Policy

- Must not overwrite a cache block unless main memory is up to date
- Multiple CPUs may have individual caches
- I/O may address main memory directly



Write Through

- All writes go to main memory as well as cache
- Multiple CPUs can monitor main memory traffic to keep local (to CPU) cache up to date
- Lots of traffic
- Slows down writes



Write Back

- Updates initially made in cache only
- Update bit for cache slot is set when update occurs
- If block is to be replaced, write to main memory only if update bit is set
- Other caches get out of sync
- I/O must access main memory through cache
- 15% of memory references are writes



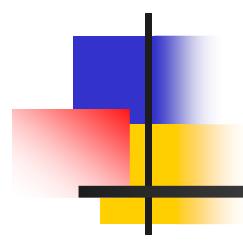
Hit Ratios and Effective Access Times

- Hit ratio and effective access time for single level cache:

$$\text{Hit ratio} = \frac{\text{No. times referenced words are in cache}}{\text{Total number of memory accesses}}$$

$$\text{Eff. access time} = \frac{(\# \text{ hits})(\text{Time per hit}) + (\# \text{ misses})(\text{Time per miss})}{\text{Total number of memory access}}$$

- Hit ratios and effective access time for multi-level cache adds another hit percentage in the above formula.



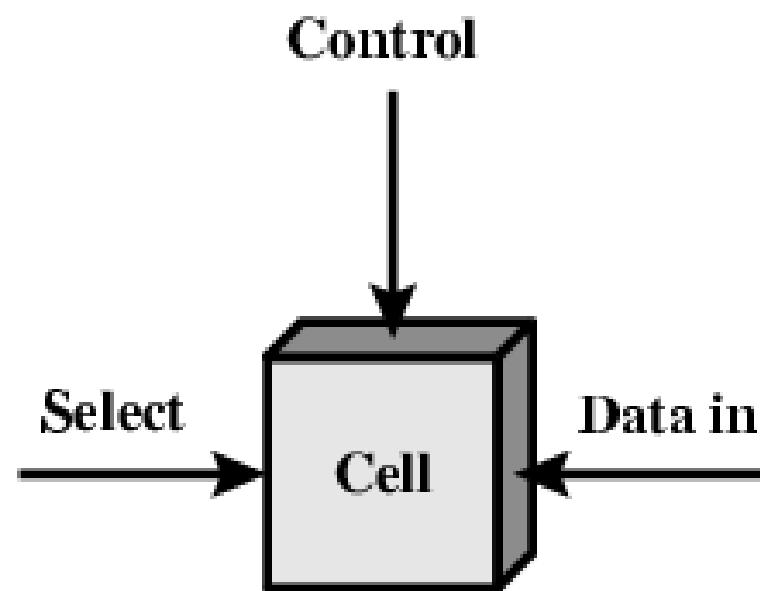
RAM



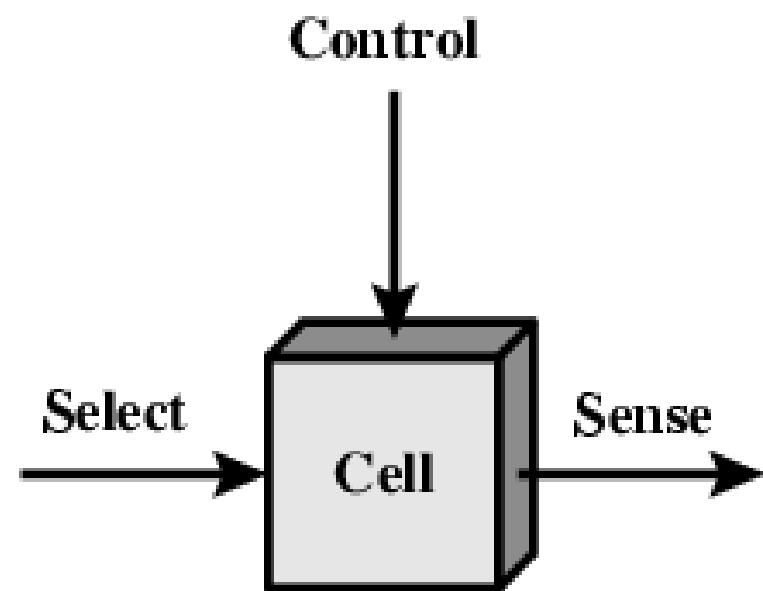
RAM

- Misnamed as every semiconductor memory is random access
- Read/Write
- Volatile
- Temporary storage
- Static or dynamic

Memory Cell Operation

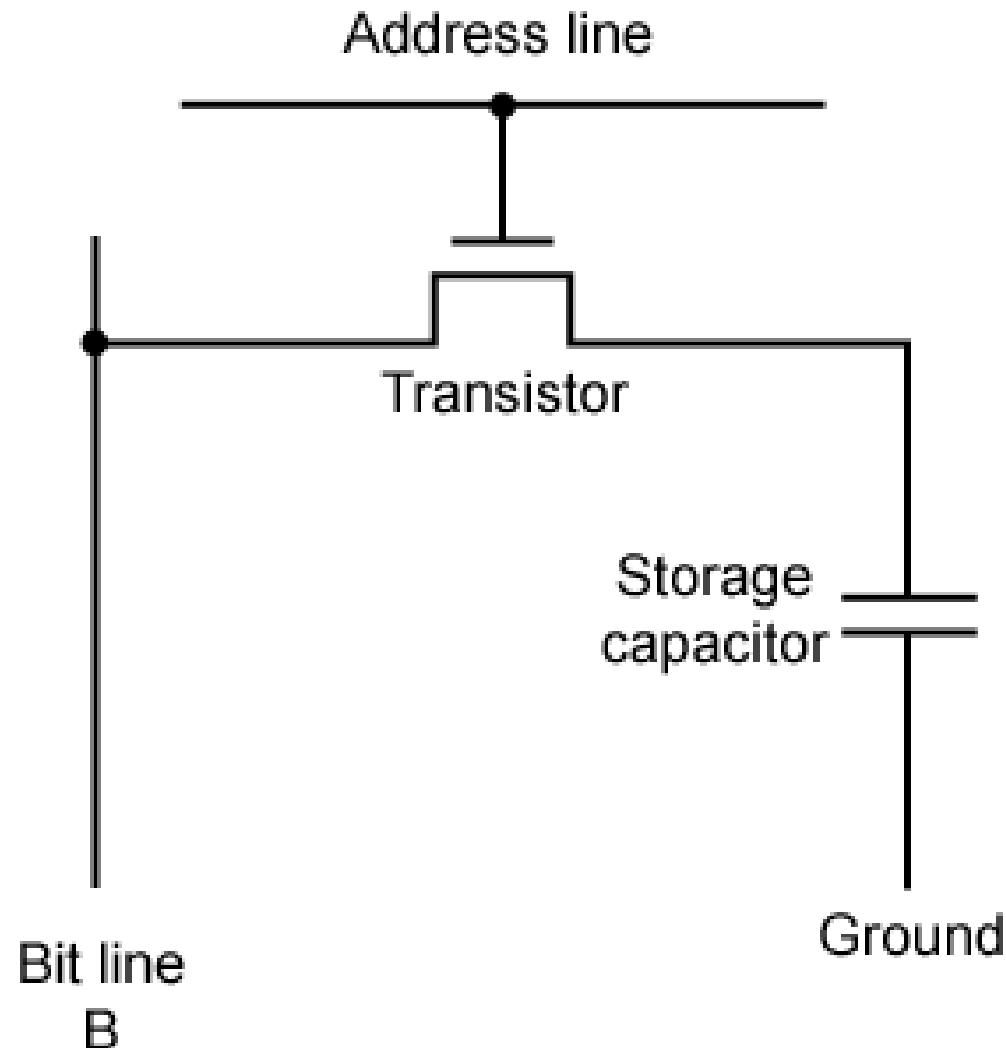


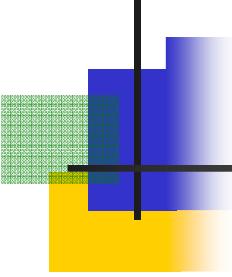
(a) Write



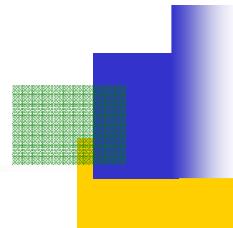
(b) Read

Dynamic RAM Structure

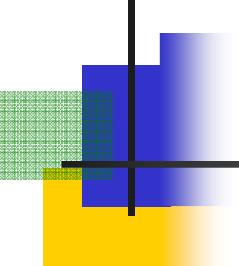




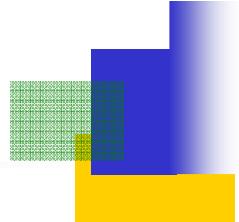
DRAM Operation



- Address line active when bit read or written
 - Transistor switch closed (current flows)
- Write
 - Voltage to bit line
 - High for 1 low for 0
 - Then signal address line
 - Transfers charge to capacitor
- Read
 - Address line selected
 - transistor turns on
 - Charge from capacitor fed via bit line to sense amplifier
 - Compares with reference value to determine 0 or 1
 - Capacitor charge must be restored

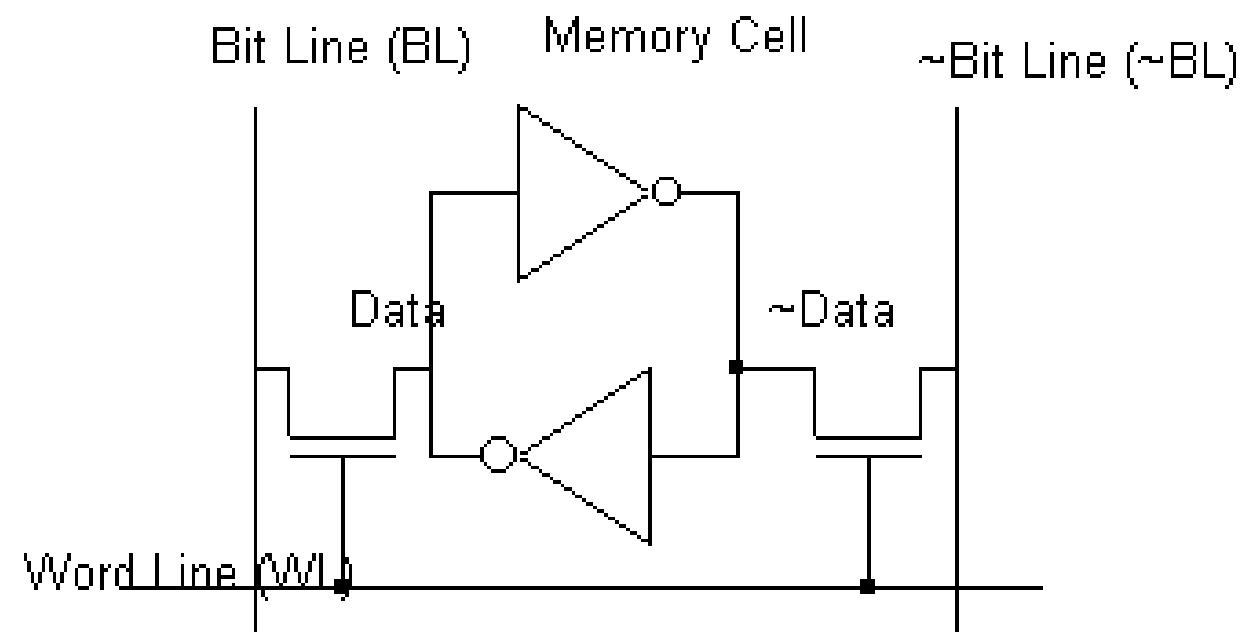


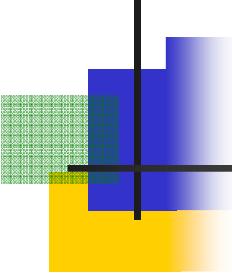
Static RAM



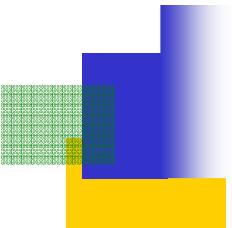
- Bits stored as on/off switches
- No charges to leak
- No refreshing needed when powered
- More complex construction
- Larger per bit
- More expensive
- Does not need refresh circuits
- Faster
- Cache
- Digital
 - Uses flip-flops

Static RAM



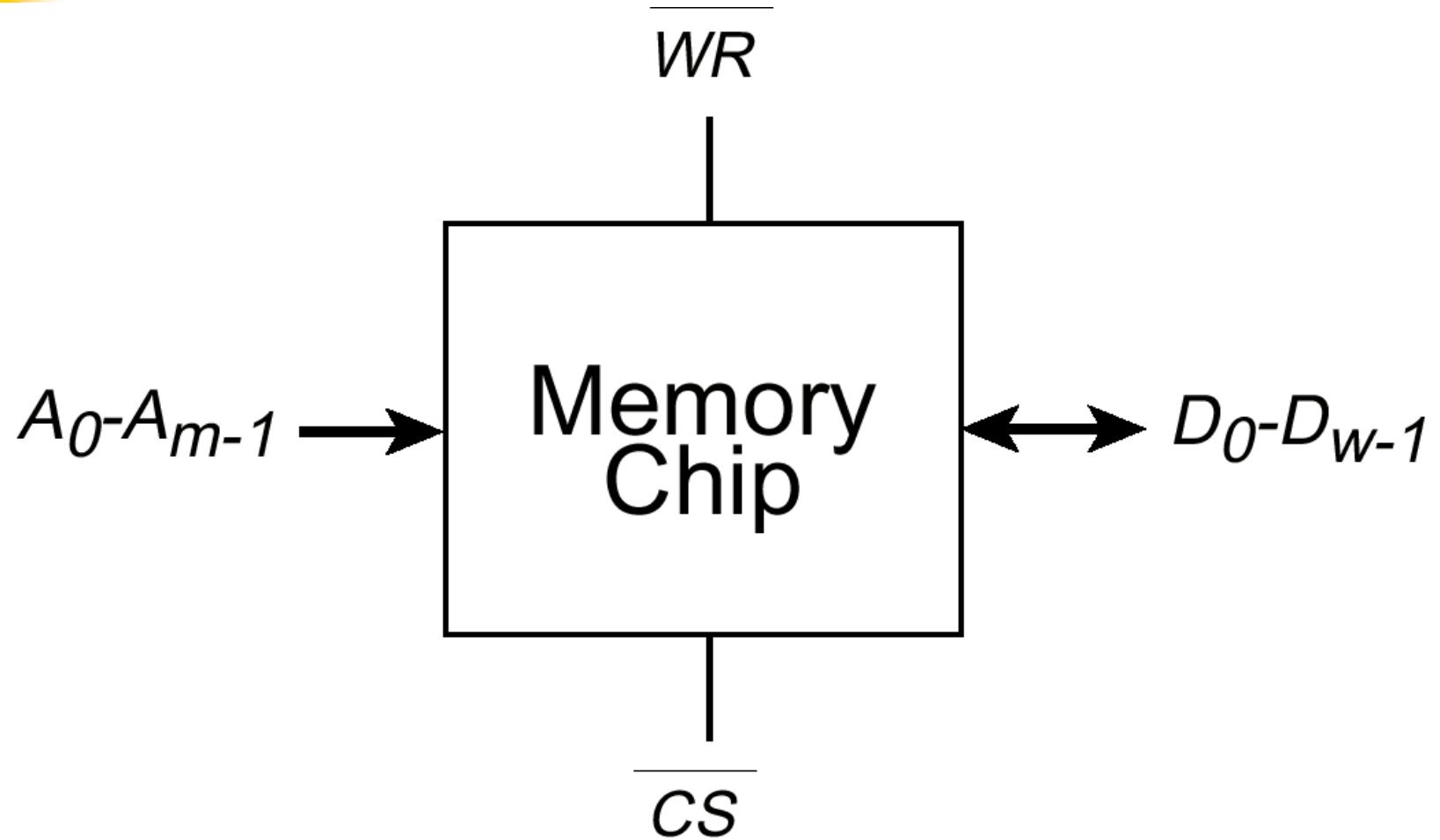


SRAM v DRAM



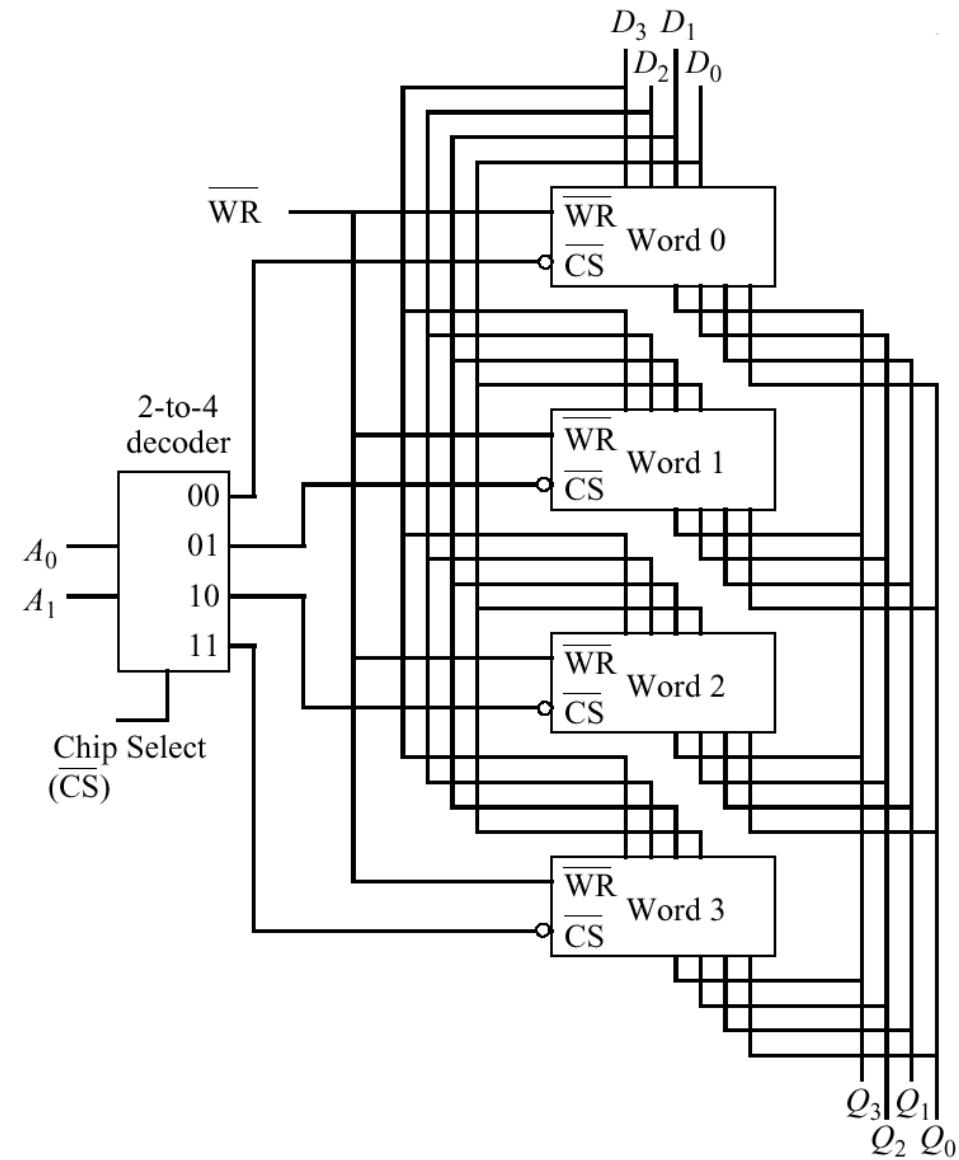
- Both volatile
 - Power needed to preserve data
- Dynamic cell
 - Simpler to build, smaller
 - More dense
 - Less expensive
 - Needs refresh
 - Larger memory units
- Static
 - Faster
 - Cache

Simplified RAM Chip Pinout

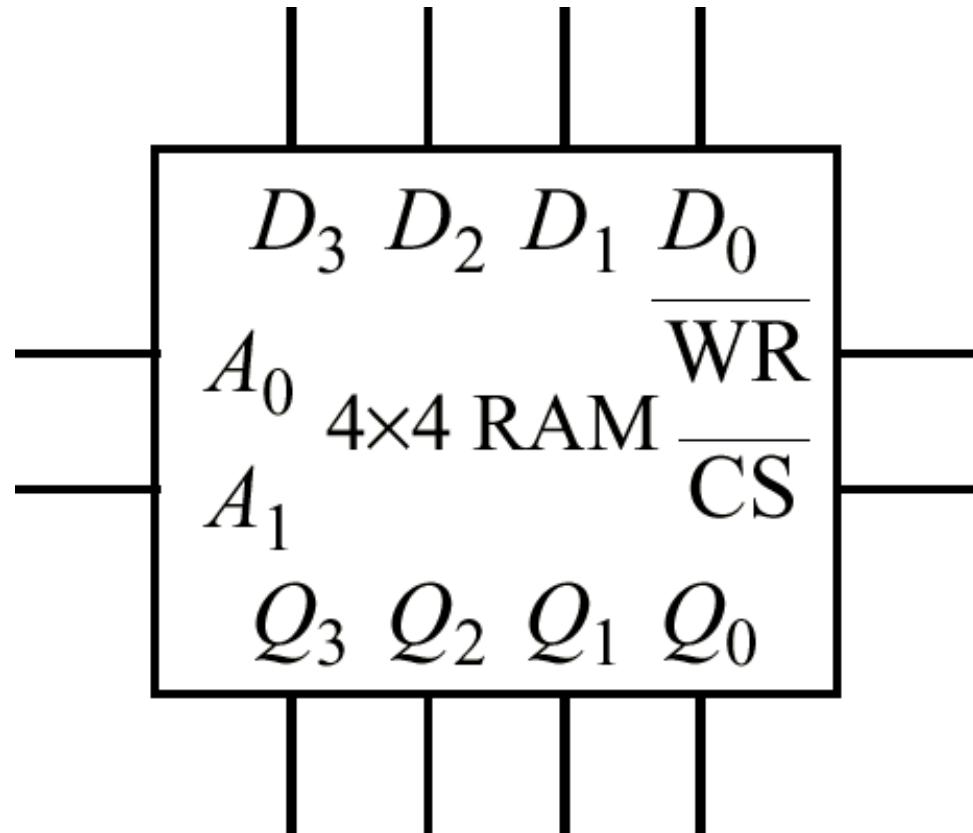


Simplified RAM Chip Pinout

A Four-Word
Memory with Four
Bits per Word in a
2D Organization

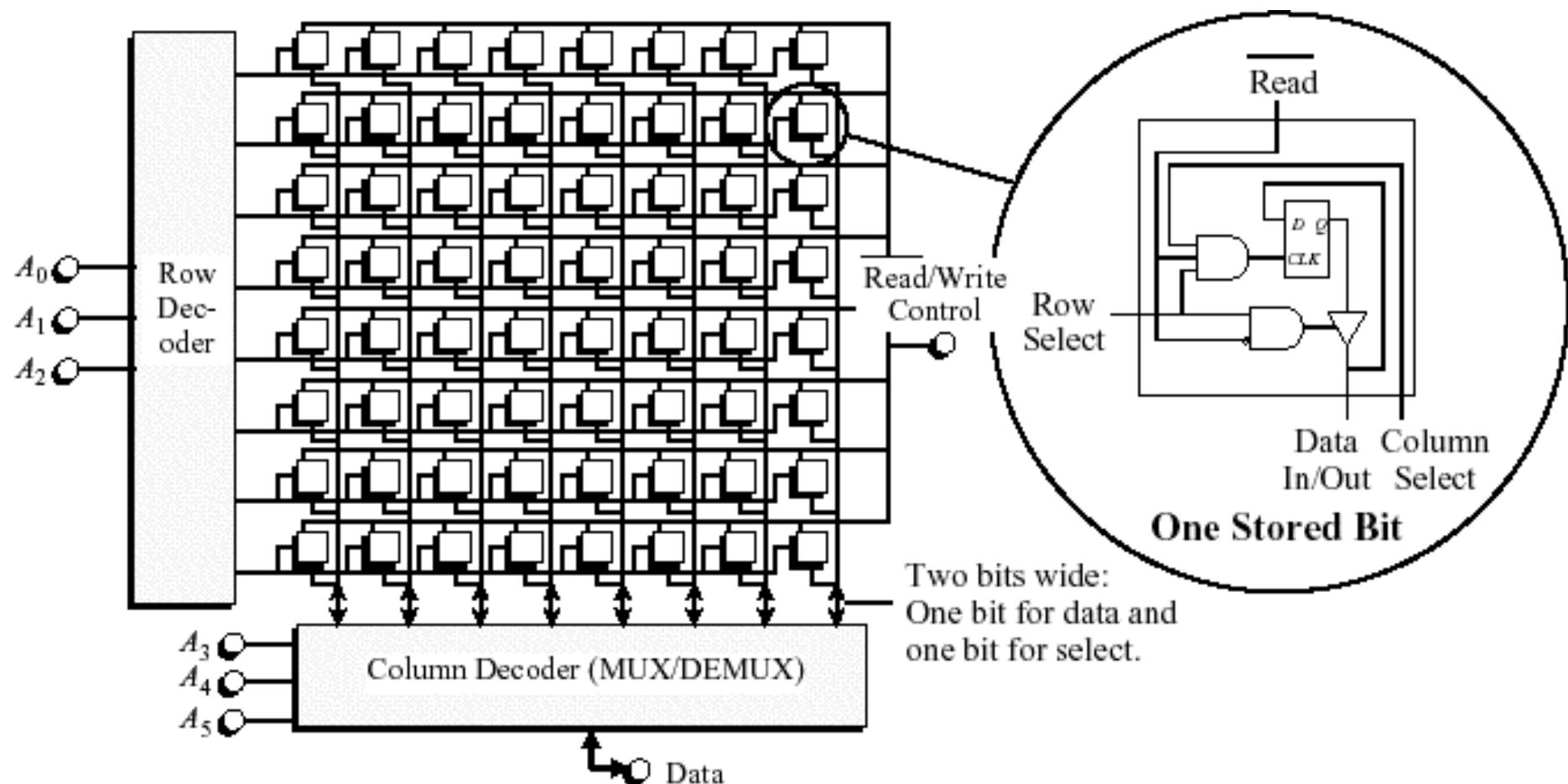


Simplified RAM Chip Pinout

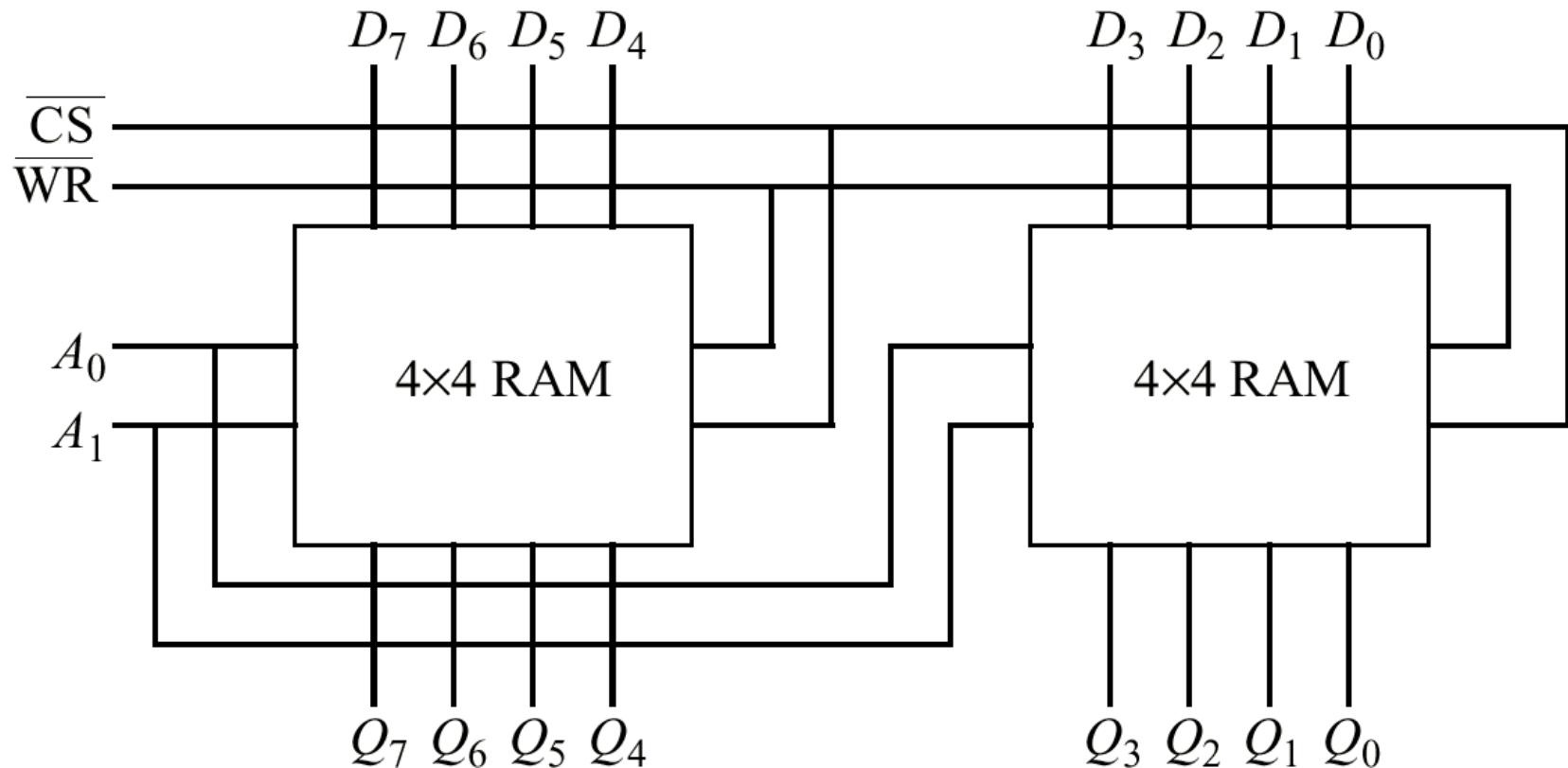


A Simplified Representation of the Four-Word by Four-Bit RAM

2 Organization of a 64-Word by 1-Bit RAM



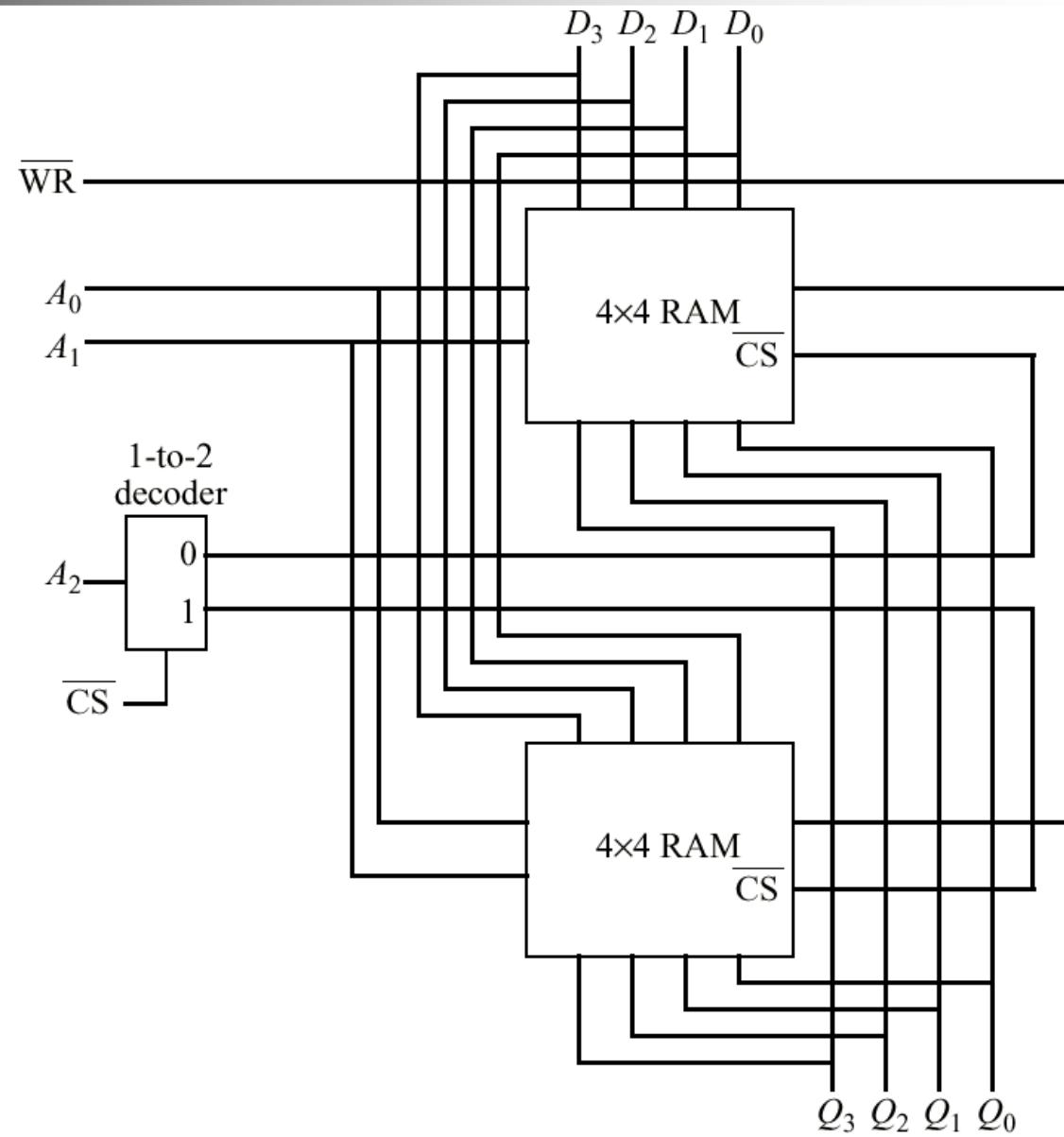
Combination of Smaller RAM Modules

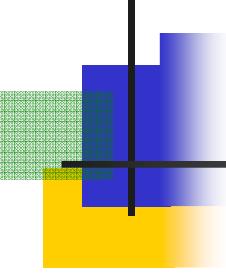


Two Four-Word by Four-Bit RAMs are Used in
Creating a Four-Word by Eight-Bit RAM

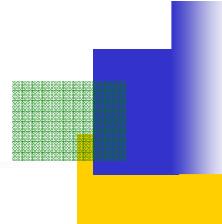
Combination of Smaller RAM Modules

Two Four-Word
by Four-Bit RAMs
Make up an
Eight-Word by
Four-Bit RAM

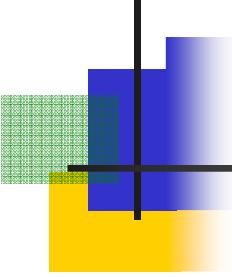




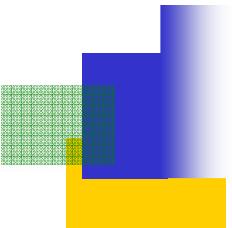
External Storage



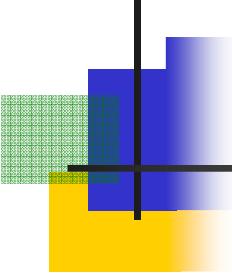
- Hard Disk
- RAID
- CD ROM
- Magnetic Tapes



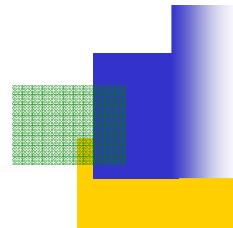
RAID



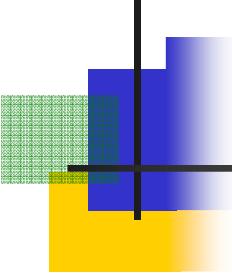
- Redundant Array of Independent Disks
- 6 levels in common use
- Not a hierarchy
- Set of physical disks viewed as single logical drive by O/S
- Data distributed across physical drives
- Can use redundant capacity to store parity information



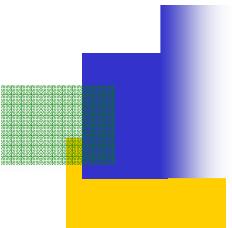
RAID 0



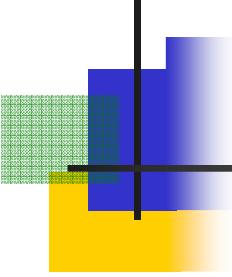
- No redundancy
- Data striped across all disks
- Round Robin striping
- Increase speed
 - Multiple data requests probably not on same disk
 - Disks seek in parallel
 - A set of data is likely to be striped across multiple disks



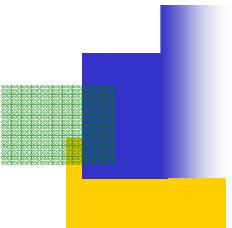
RAID 1



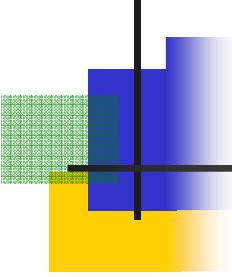
- Mirrored Disks
- Data is striped across disks
- 2 copies of each stripe on separate disks
- Read from either
- Write to both
- Recovery is simple
 - Swap faulty disk & re-mirror
 - No down time
- Expensive



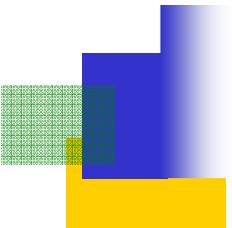
RAID 2



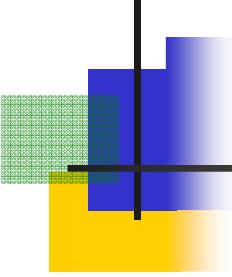
- Disks are synchronized
- Very small stripes
 - Often single byte/word
- Error correction calculated across corresponding bits on disks
- Multiple parity disks store Hamming code error correction in corresponding positions
- Lots of redundancy
 - Expensive
 - Not used



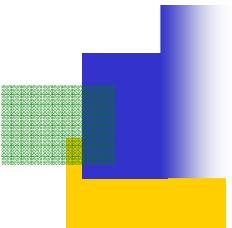
RAID 3



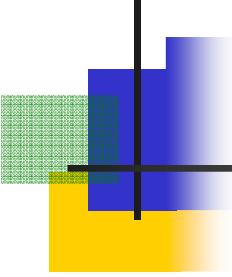
- Similar to RAID 2
- Only one redundant disk, no matter how large the array
- Simple parity bit for each set of corresponding bits
- Data on failed drive can be reconstructed from surviving data and parity info
- Very high transfer rates



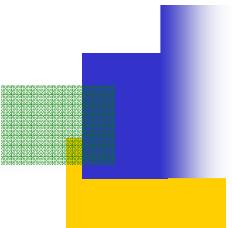
RAID 4



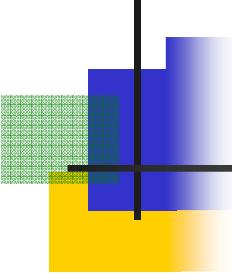
- Each disk operates independently
- Good for high I/O request rate
- Large stripes
- Bit by bit parity calculated across stripes on each disk
- Parity stored on parity disk



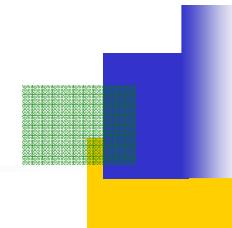
RAID 5



- Like RAID 4
- Parity striped across all disks
- Round robin allocation for parity stripe
- Avoids RAID 4 bottleneck at parity disk
- Commonly used in network servers

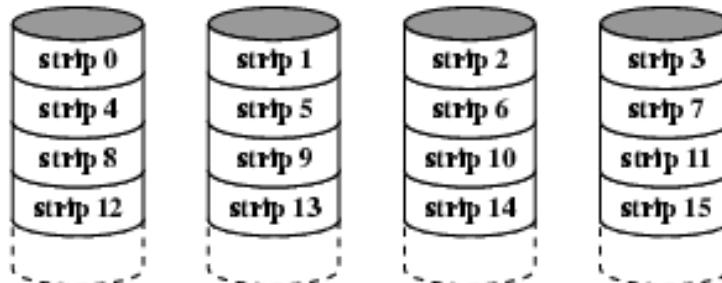


RAID 6

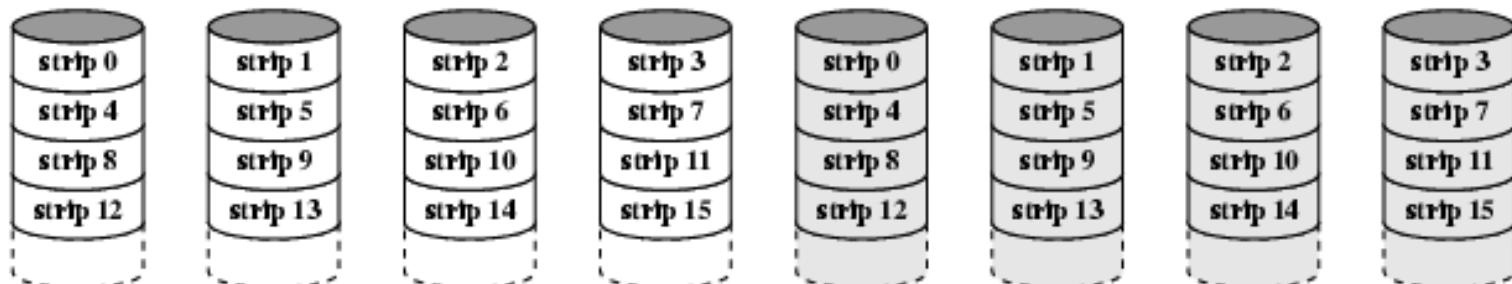


- Two parity calculations
- Stored in separate blocks on different disks
- User requirement of N disks needs $N+2$
- High data availability
 - Three disks need to fail for data loss
 - Significant write penalty

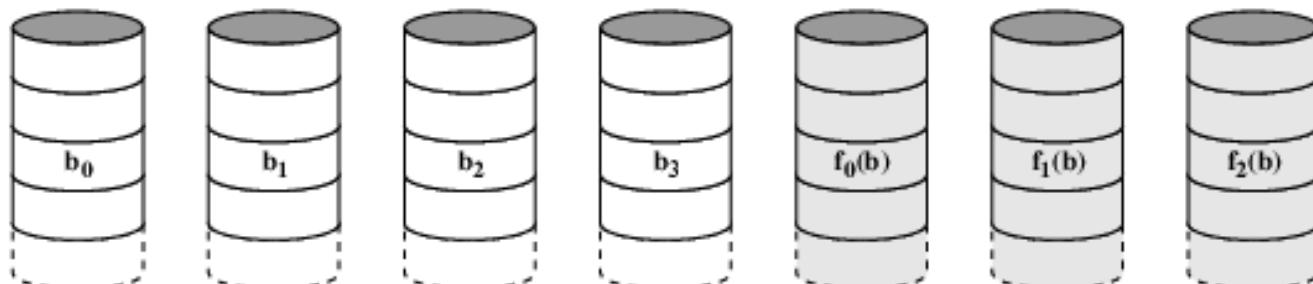
RAID 0, 1, 2



(a) RAID 0 (non-redundant)

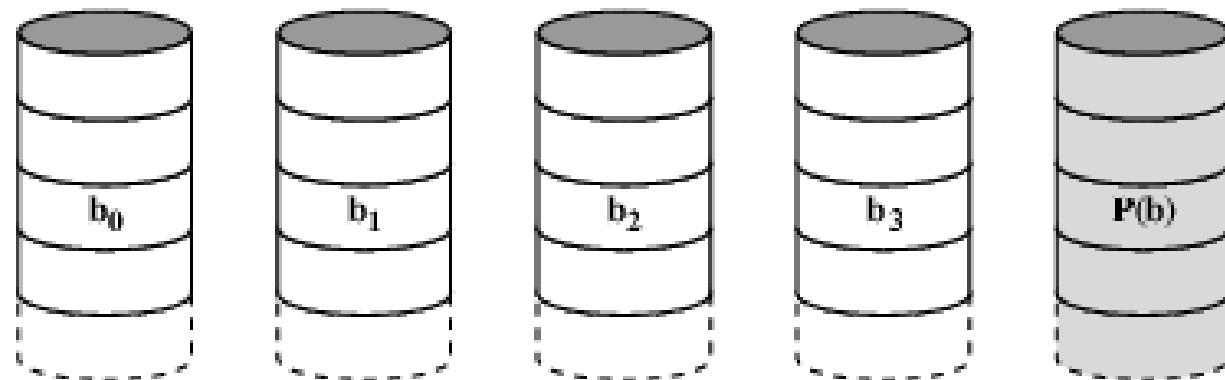


(b) RAID 1 (mirrored)

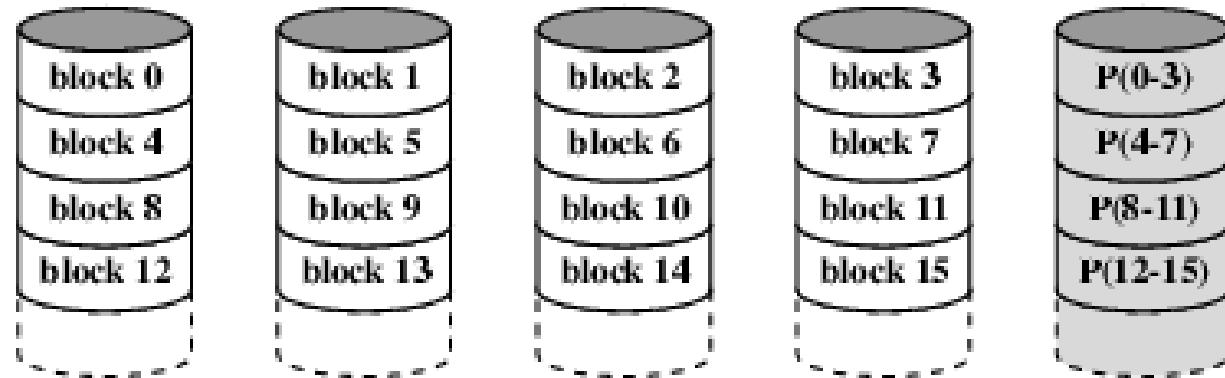


(c) RAID 2 (redundancy through Hamming code)

RAID 3 & 4

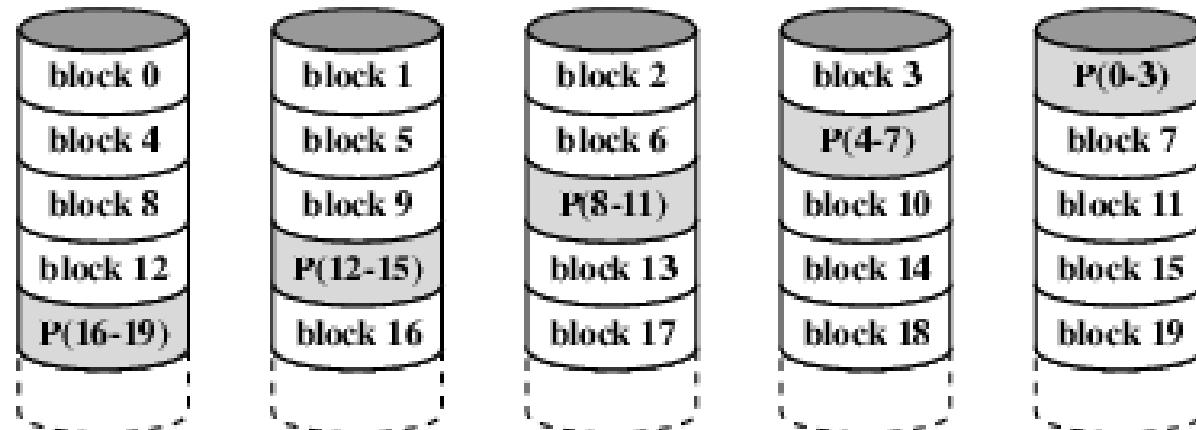


(d) RAID 3 (bit-interleaved parity)

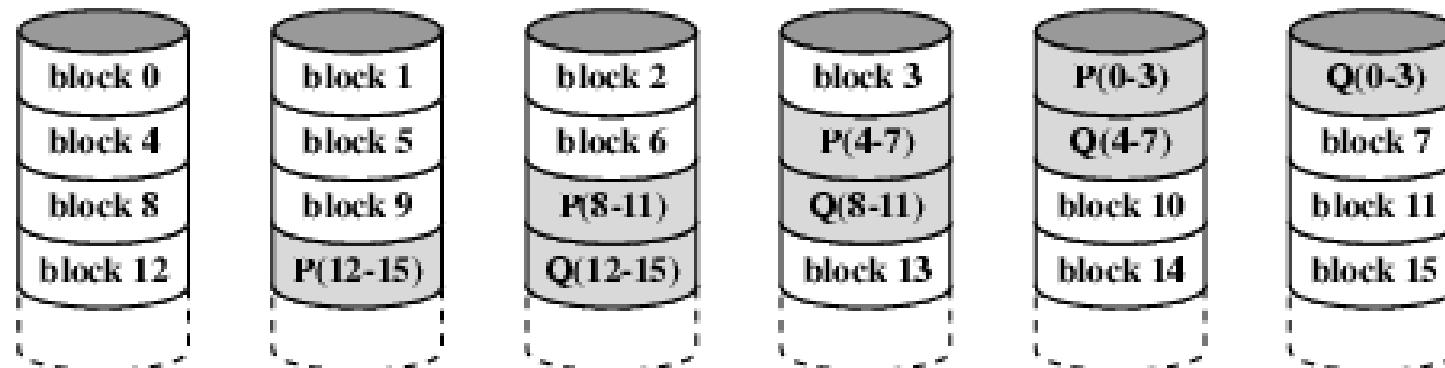


(e) RAID 4 (block-level parity)

RAID 5 & 6



(f) RAID 5 (block-level distributed parity)

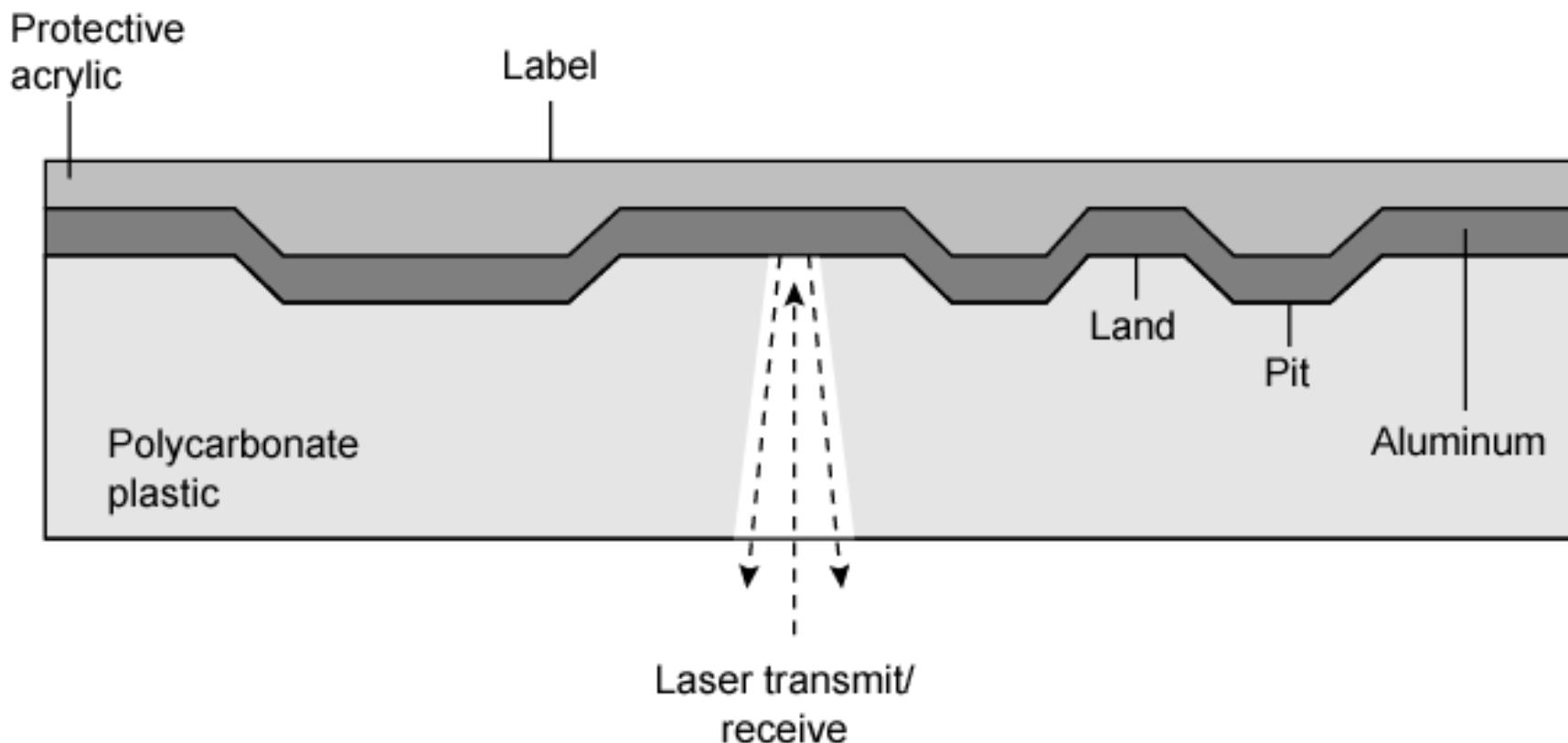


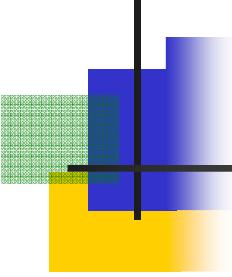
(g) RAID 6 (dual redundancy)

Optical Storage CD-ROM

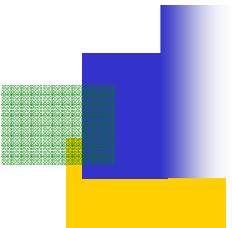
- Originally for audio
- 650Mbytes giving over 70 minutes audio
- Polycarbonate coated with highly reflective coat, usually aluminium
- Data stored as pits
- Read by reflecting laser
- Constant packing density
- Constant linear velocity

CD Operation



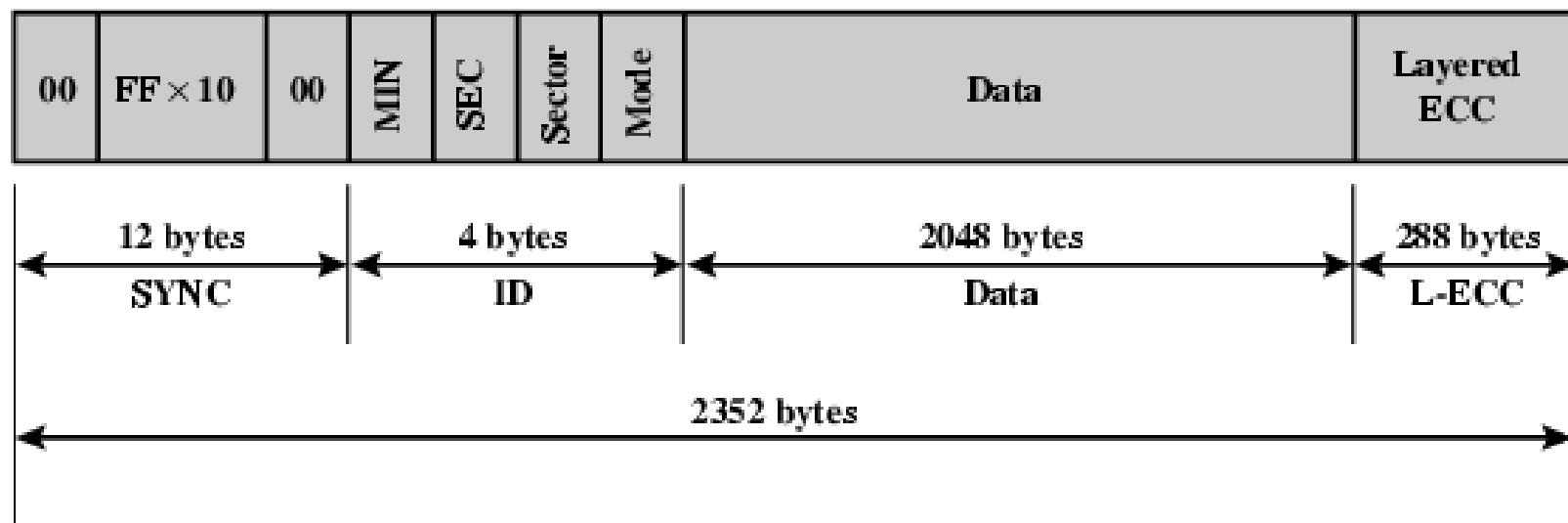


CD-ROM Drive Speeds



- Audio is single speed
 - Constant linear velocity
 - 1.2 ms^{-1}
 - Track (spiral) is 5.27km long
 - Gives 4391 seconds = 73.2 minutes
- Other speeds are quoted as multiples
- e.g. 24x
- Quoted figure is maximum drive can achieve

CD-ROM Format



- Mode 0=blank data field
- Mode 1=2048 byte data + error correction
- Mode 2=2336 byte data



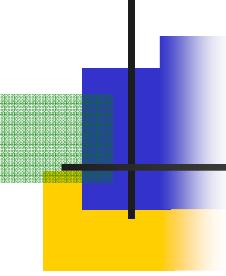
Random Access on CD-ROM

- Difficult
- Move head to rough position
- Set correct speed
- Read address
- Adjust to required location

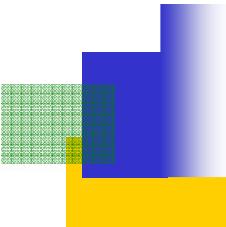


Other Optical Storage

- CD-Recordable (CD-R)
 - Compatible with CD-ROM drives
- CD-RW
 - Erasable
 - Getting cheaper
 - Mostly CD-ROM drive compatible



DVD - what's in a name?



- Digital Video Disk
 - Used to indicate a player for movies
 - Only plays video disks
- Digital Versatile Disk
 - Used to indicate a computer drive
 - Will read computer disks and play video disks



DVD - technology

- Multi-layer
- Very high capacity (4.7G per layer)
- Almost the same speed as CD ROM



Queries?